

EL764880394 US

Christopher M. Connors; Andrew F. Miller; Joshua P. Walsky;  
Jeffrey R. Van Dyke; James Singh; Andrew Leamon

5

This application is a continuation-in-part of Patent Application No. (\_\_\_ Atty. Docket No. M-9723 US \_\_\_), entitled "METHOD AND APPARATUS FOR SORTING PRODUCTS BY FEATURES," filed 1/12/01 and having C. Connors, A. Miller, J. Walsky, J. Van Dyke, J. Singh and A. Leamon as inventors (which claims priority to Provisional Patent Application No. 60/176,117, entitled "SYSTEM AND METHOD FOR FACILITATING COMMERCIAL TRANSACTIONS OVER A DATA NETWORK," filed 1/14/00 and having J. Walsky and S. Aboel-Nil as inventors); and is related to Patent Application Nos. (\_\_\_ Atty. Docket No. M-9722 US \_\_\_) (filed 1/12/01) and (\_\_\_ Atty. Docket No. M-9722-1P US \_\_\_) (filed herewith), both entitled "METHOD AND APPARATUS FOR PRODUCT COMPARISON," and both having C. Connors, A. Miller, J. Walsky, J. Van Dyke, J. Singh and A. Leamon as inventors, and Patent Application Nos. (\_\_\_ Atty. Docket No. M-9724 US \_\_\_) (filed 1/12/01) and (\_\_\_ Atty. Docket No. M-9724-1P US \_\_\_) (filed herewith), both entitled "SYSTEM AND METHOD FOR FACILITATING COMMERCIAL TRANSACTIONS OVER A DATA NETWORK," and both having J. Walsky as inventor. These applications are assigned to Trilog Development Group, Inc., the assignee of the present invention, and are hereby incorporated by reference, in their entirety and for all purposes.

## **BACKGROUND OF THE INVENTION**

### **Field of the Invention**

The present invention relates to transacting commerce over a network, and, more particularly, to a method and apparatus for processing information related to such commercial transactions.

### **Description of the Related Art**

Historically, information regarding commercially available products has been disseminated using various types of well-known media, including print, radio, television and the like. The operation of such media has been patently obvious, and its use self-evident. With the advent of the Internet, a wide array of product information has become even more accessible to the average person.

Entities providing such product information have been assisted in their endeavor by networking and client/server technology that has become available in approximately the last ten to fifteen years. Such client/server arrangements typically allow a number of users employing client terminals to communicate with a remote server computer in order to transfer information therebetween. This information may include text, data in any one of a number of formats, graphical information, streaming audio and video, and other such information. To facilitate such transfers, client terminals can employ a "web" browser that provides access to a server via a graphical user interface (GUI). The server responds to requests from the client by providing information in the form of a "web page." One popular collection of servers uses Hypertext Transfer Protocol (HTTP) to provide information. This assemblage is known as the "World Wide Web" (WWW). A collection of related web pages is often referred to as a "website," or more simply a "site." The information is typically presented as web pages written as text with standardized formatting and control symbols known as Hypertext Mark-up Language (HTML). HTML provides basic hypertext document formatting and allows a server to specify "links" to other servers and files. Use of an HTML-compliant browser involves specification of a link via a Uniform Resource Locator (URL). Upon such specification, the user's client terminal

makes a TCP/IP request to the server identified in the link and receives an HTML file that is interpreted by the browser so that an electronic HTML document made up of one or more web pages may be displayed on the client's terminal.

Unfortunately, however, the use of such functionality for the selection and purchase of products is less than ideal. This is especially true in the case where comparable products exist. Especially when faced with a large number of comparable products, it is natural for a consumer to want to compare such products (e.g., in side-by-side fashion), in order to better understand the differences between the products, and the advantages of each. Moreover, when making such comparisons, and when employing web-based technology in general, reducing the need for user interaction can improve a website's ability to handle increased traffic, and ultimately, the consumer's buying experience.

#### **SUMMARY OF THE INVENTION**

In one embodiment of the present invention, a software architecture is disclosed. The software architecture includes a database layer, a services layer (coupled to said database layer) and a needs analysis module (coupled to said services layer).

In various aspects of this embodiment, the software architecture further includes the following features. For example, the needs analysis module can be configured to permit identification of a product based on attribute information. Moreover, the services layer can include a filter service. The filter service can be configured to provide a product identifier to the needs analysis module in response to a product attribute received from the needs analysis module, where the product identifier identifies a product, and the product attribute is an attribute of the product. Similarly, the database layer can include a database, with the filter service being configured to use the product attribute to retrieve the product identifier from the database.

In another aspect of this embodiment, the software architecture's needs analysis module is configured to permit identification of a product configuration based on product identifier information. To achieve this end, the services layer can include a configuration service. The product identifier identifies a product. The configuration service can be configured to provide a configuration list to the needs analysis module in response to a product identifier received from the needs analysis module.

In one embodiment of the present invention, a software architecture is disclosed. The software architecture includes a database layer and a services layer. The services layer is coupled to the database layer and includes a filter service.

In various aspects of this embodiment, the software architecture further includes the following features. For example, the filter service can be configured to permit identification of a product based on attribute information. Moreover, the software architecture can further include a module layer coupled to the services layer, which includes a needs analysis module. The filter service can be configured to provide a product identifier to the needs analysis module in response to a product attribute received from the needs analysis module, with the product identifier identifying a product and the product attribute being an attribute of the product. Similarly, the database layer can include a database, with the filter service configured to use the product attribute to retrieve the product identifier from the database.

In one embodiment of the present invention, a software architecture is disclosed. The software architecture includes a database layer and a services layer. The services layer is coupled to the database layer and includes a configuration service.

In various aspects of this embodiment, the software architecture further includes the following features. For example, the needs analysis module can be configured to permit identification of a product configuration based on product identifier information. Similarly, the configuration service can be configured to permit identification of a product based on a product identifier.

In another aspect of this embodiment, the software architecture includes a module layer that is coupled to the services layer. The module layer can include a needs analysis module. In a further aspect, the configuration service is configured to provide a configuration list to the needs analysis module in response to a product identifier received from the needs analysis module, where the product identifier identifies a product.

In one embodiment of the present invention, a method for identifying a product is disclosed. The method includes selecting a selected feature from a number of features, determining which of the products is configured with the selected feature and identifying the product as being configured with the selected feature. In this embodiment, the product is one of a number of products, the product is configured with the selected feature, and each of the products is configured with at least one of the features. In one aspect of this embodiment, the selected feature is one of a number of selected features, the selected features form a product configuration, and the product configuration is an allowable product configuration.

The foregoing is a summary and thus contains, by necessity, simplifications, generalizations and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. As will also be apparent to one of skill in the art, the operations disclosed herein may be implemented in a number of ways, and such changes and modifications may be made without departing from this invention and its broader aspects. Other aspects, inventive features, and advantages of the present invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Fig. 1 is a block diagram illustrating a network environment in which commercial transaction processing according to embodiments of the present invention may be practiced.

Fig. 2 depicts a block diagram of a computer system suitable for implementing  
5 embodiments of the present invention.

Fig. 3 depicts the interconnection of the computer system of Fig. 2 to client and host systems.

Fig. 4 is a plan view of a web page having a visual representation of a hypertext link and data entry regions employed to access the present invention.

10 Fig. 5 is a plan view of a web page employed to allow users to register in order to gain access to the present invention.

Fig. 6 is a plan view of a web page that is uniquely associated with a user and on which product-related information stored on a server is accessed, in accordance with the present invention.

15 Fig. 7 is a plan view of a web page through which a user can access product-related information according to embodiments of the present invention.

Fig. 8A is a plan view of a web page for entering information regarding a user's location according to embodiments of the present invention.

20 Fig. 8B is a plan view of a web page for entering information regarding a desired manufacturer according to embodiments of the present invention.

Fig. 9 is a plan view of a web page through which a user can access product-related information for selected products according to embodiments of the present invention.

Fig. 10 is a plan view of a web page through which a user can access product-related information for a selected product according to embodiments of the present invention.

Fig. 11 is a plan view of a page through which a user can access product-related performance information for a selected product according to embodiments of the present invention.

Fig. 12 is a plan view of a web page through which a user can access product-related feature information for a selected product according to embodiments of the present invention.

Fig. 13 is a plan view of a web page that allows a user to identify a desired product by price range according to embodiments of the present invention.

Fig. 14 is a plan view of a web page that allows a user to identify a desired product by vehicle type according to embodiments of the present invention.

Fig. 15 is a plan view of a web page that allows a user to identify a desired product by engine characteristics according to embodiments of the present invention.

Fig. 16 is a plan view of a web page that allows a user to identify a desired product by fuel economy according to embodiments of the present invention.

Figs. 17 is a plan view of a web page that allows a user to identify a desired product by interior characteristics according to embodiments of the present invention.

Fig. 18 is a plan view of a web page that allows a user to identify a desired product by safety features according to embodiments of the present invention.

Fig. 19 is a flow diagram illustrating a product identification process according to embodiments of the present invention.

Fig. 20 is a block diagram illustrating a software architecture according to embodiments of the present invention.

Fig. 21 is a block diagram illustrating the creation of a database according to embodiments of the present invention.

Fig. 22 illustrates an example of a database structure that can be used in the database of Fig. 21.

5 Fig. 23 is a flow diagram illustrating a product identification process based on product attributes (e.g., features) that occurs in a software architecture according to embodiments of the present invention.

Fig. 24 is a flow diagram illustrating a product identification process based on a product identifier that occurs in a software architecture according to embodiments of  
10 the present invention.

Fig. 25 is a block diagram illustrating an example of a website according to embodiments of the present invention.

The use of the same reference symbols in different drawings indicates similar or identical items.

## 15 **DETAILED DESCRIPTION OF THE INVENTION**

The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather, any number of variations may fall within the scope of the invention which is defined in the claims following the description.

### 20 **Introduction**

The present invention provides a software architecture and process that allows a user the ability to identify a product (or configurations thereof) by specifying attributes desired of the product using, for example, a WWW interface such as a web browser to access the requisite information remotely (e.g., over a network such as the  
25 Internet). Using a website employing such a software architecture (also referred to herein as a website architecture), a user is able to identify allowable product



configurations (allowable combinations of features (i.e., attributes); also referred to more simply herein as configurations) based on product attributes (also referred to herein as attribute information), display the allowable product configurations thus selected, identify the features (attributes) of a given product, generate comparable product configurations, store the product configuration(s) and perform related tasks. The foregoing functionalities are provided by a software module (referred to herein as a needs analysis module) that enables the identification of products based on their features (i.e., attributes). These functionalities are enhanced by the use of a database structure that permits the retrieval of configurations and/or related attributes using a product identifier (or more simply, an identifier), a configuration, one or more product attributes or other such information. Moreover, by persisting configuration information throughout such a system, a user need not re-enter product information, thus enhancing the user's purchasing experience.

#### **An Example Computing and Network Environment**

Fig. 1 is a block diagram illustrating a network environment in which a commercial transaction processing system according to the present invention may be practiced. As is illustrated in Fig. 1, network 5, such as a private wide area network (WAN) or the Internet, includes a number of networked servers 6(1)-(N) that are accessible by client terminals 7(1)-(N). Communication between client terminals 7(1)-(N) and servers 6(1)-(N) typically occurs over a publicly accessible network, such as a public switched telephone network over ADSL telephone lines or large bandwidth trunks, for example communications channels providing T1 or OC3 service. Client terminals 7(1)-(N) access servers 6(1)-(N) through a service provider, e.g., an Internet service provider such as America On-Line <sup>TM</sup>, Prodigy <sup>TM</sup>, CompuServe <sup>TM</sup> and the like, by executing application specific software, commonly referred to as a browser, on one of client terminals 7(1)-(N).

One or more of client terminals 7(1)-(N) and/or one or more of servers 6(1)-(N) may be, for example, a computer system of any appropriate design, in general, including a mainframe, a mini-computer or a personal computer system. Such a

computer system typically includes a system unit having a system processor and associated volatile and non-volatile memory, one or more display monitors and keyboards, one or more diskette drives, one or more fixed disk storage devices and one or more printers. These computer systems are typically information handling systems which are designed to provide computing power to one or more users, either locally or remotely. Such a computer system may also include one or a plurality of I/O devices (i.e. peripheral devices) which are coupled to the system processor and which perform specialized functions. Examples of I/O devices include modems, sound and video devices and specialized communication devices. Mass storage devices such as hard disks, CD-ROM drives and magneto-optical drives may also be provided, either as an integrated or peripheral device. One such example computer system, discussed in terms of client terminals 7(1)-(N) is shown in detail in Fig. 2.

Fig. 2 depicts a block diagram of a computer system 10 suitable for implementing the present invention, and example of one or more of client terminals 7(1)-(N). Computer system 10 includes a bus 12 which interconnects major subsystems of computer system 10 such as a central processor 14, a system memory 16 (typically RAM, but which may also include ROM, flash RAM, or the like), an input/output controller 18, an external audio device such as a speaker system 20 via an audio output interface 22, an external device such as a display screen 24 via display adapter 26, serial ports 28 and 30, a keyboard 32 (interfaced with a keyboard controller 33), a storage interface 34, a floppy disk drive 36 operative to receive a floppy disk 38, and a CD-ROM drive 40 operative to receive a CD-ROM 42. Also included are a mouse 46 (or other point-and-click device, coupled to bus 12 via serial port 28), a modem 47 (coupled to bus 12 via serial port 30) and a network interface 48 (coupled directly to bus 12).

Bus 12 allows data communication between central processor 14 and system memory 16, which may include both read only memory (ROM) or flash memory (neither shown), and random access memory (RAM) (not shown), as previously noted. The RAM is generally the main memory into which the operating system and application programs are loaded and typically affords at least 16 megabytes of

memory space. The ROM or flash memory may contain, among other code, the Basic Input-Output system (BIOS) which controls basic hardware operation such as the interaction with peripheral components. Applications resident with computer system 10 are generally stored on and accessed via a computer readable medium, such as a hard disk drive (e.g., fixed disk 44), an optical drive (e.g., CD-ROM drive 40), floppy disk unit 36 or other storage medium. Additionally, applications may be in the form of electronic signals modulated in accordance with the application and data communication technology when accessed via network modem 47 or interface 48.

Storage interface 34, as with the other storage interfaces of computer system 10, may connect to a standard computer readable medium for storage and/or retrieval of information, such as a fixed disk drive 44. Fixed disk drive 44 may be a part of computer system 10 or may be separate and accessed through other interface systems. Many other devices can be connected such as a mouse 46 connected to bus 12 via serial port 28, a modem 47 connected to bus 12 via serial port 30 and a network interface 48 connected directly to bus 12. Modem 47 may provide a direct connection to a remote server via a telephone link or to the Internet via an internet service provider (ISP). Network interface 48 may provide a direct connection to a remote server via a direct network link to the Internet via a POP (point of presence). Network interface 48 may provide such connection using wireless techniques, including digital cellular telephone connection, Cellular Digital Packet Data (CDPD) connection, digital satellite data connection or the like.

Many other devices or subsystems (not shown) may be connected in a similar manner (e.g., bar code readers, document scanners, digital cameras and so on). Conversely, it is not necessary for all of the devices shown in Fig. 2 to be present to practice the present invention. The devices and subsystems may be interconnected in different ways from that shown in Fig. 2. The operation of a computer system such as that shown in Fig. 2 is readily known in the art and is not discussed in detail in this application. Code to implement the present invention may be operably disposed or stored in computer-readable storage media such as one or more of system memory 16, fixed disk 44, CD-ROM 42, or floppy disk 38. Additionally, computer system 10 may

be any kind of computing device, and so includes personal data assistants (PDAs), network appliance, X-window terminal or other such computing device. The operating system provided on computer system 10 may be MS-DOS®, MS-WINDOWS®, OS/2®, UNIX®, Linux® or other known operating system. Computer  
 5 system 10 also supports a number of Internet access tools, including, for example, an HTTP-compliant web browser having a JavaScript interpreter, such as Netscape Navigator® 3.0, Microsoft Explorer® 3.0 and the like.

It will be noted that the variable identifier "N" is used in several instances in Fig. 2 to more simply designate the final element (e.g., servers 6(1)-(N) and client  
 10 terminals 7(1)-(N)) of a series of related or similar elements (e.g., servers and client terminals). The repeated use of such variable identifiers is not meant to imply a correlation between the sizes of such series of elements, although such correlation may exist. The use of such variable identifiers does not require that each series of  
 15 elements has the same number of elements as another series delimited by the same variable identifier. Rather, in each instance of use, the variable identified by "N" may hold the same or a different value than other instances of the same variable identifier.

Moreover, regarding the signals described herein, those skilled in the art will recognize that a signal may be directly transmitted from a first block to a second block, or a signal may be modified (e.g., amplified, attenuated, delayed, latched,  
 20 buffered, inverted, filtered or otherwise modified) between the blocks. Although the signals of the above described embodiment are characterized as transmitted from one block to the next, other embodiments of the present invention may include modified signals in place of such directly transmitted signals as long as the informational and/or functional aspect of the signal is transmitted between blocks. To some extent, a signal  
 25 input at a second block may be conceptualized as a second signal derived from a first signal output from a first block due to physical limitations of the circuitry involved (e.g., there will inevitably be some attenuation and delay). Therefore, as used herein, a second signal derived from a first signal includes the first signal or any modifications to the first signal, whether due to circuit limitations or due to passage

through other circuit elements which do not change the informational and/or final functional aspect of the first signal.

The foregoing described embodiment wherein the different components are contained within different other components (e.g., the various elements shown as components of computer system 10). It is to be understood that such depicted architectures are merely examples, and that in fact many other architectures can be implemented which achieve the same functionality. In an abstract, but still definite sense, any arrangement of components to achieve the same functionality is effectively "associated" such that the desired functionality is achieved. Hence, any two components herein combined to achieve a particular functionality can be seen as "associated with" each other such that the desired functionality is achieved, irrespective of architectures or intermediate components. Likewise, any two components so associated can also be viewed as being "operably connected", or "operably coupled", to each other to achieve the desired functionality.

Fig. 3 is a block diagram depicting a network 50 in which computer system 10 is coupled to an internetwork 51, which is coupled, in turn, to client systems 52 and 53, as well as a server 54. Internetwork 51 (e.g., the Internet) is also capable of coupling client systems 52 and 53, and server 54 to one another. With reference to computer system 10, modem 47, network interface 48 or some other method can be used to provide connectivity from computer system 10 to internetwork 51. Computer system 10, client system 52 and client system 53 are able to access information on server 54 using, for example, a web browser (not shown). Such a web browser allows computer system 10, as well as client systems 52 and 53, to access data on server 54 representing the pages of a website hosted on server 54. Protocols for exchanging data via the Internet are well known to those skilled in the art. Although Fig. 3 depicts the use of the Internet for exchanging data, the present invention is not limited to the Internet or any particular network-based environment.

Referring to Figs. 1, 2 and 3, a browser running on computer system 10 employs a TCP/IP connection to pass a request to server 54, which can run an HTTP

"service" (e.g., under the WINDOWS® operating system) or a "daemon" (e.g., under the UNIX® operating system), for example. Such a request can be processed, for example, by contacting an HTTP server 10a at the following address "http://www.carOrder.com," employing a protocol that can be used to communicate  
 5 between the server 10a and the client terminal 12. The HTTP server 10 then responds to the protocol, typically by sending a "web page" formatted as an HTML file. The browser interprets the HTML file and may form a visual representation of the same using local resources, e.g., fonts and colors.

Referring to Figs. 1, 2 and 3, server 54 (which includes an HTTP server, as  
 10 previously noted) can further include a web server (not shown), an application server (also not shown) and a database server operating with an open distributed object computing infrastructure such as Common Object Request Broker Architecture (CORBA) (also not shown). The web server can include a user interface layer and a centralized dispatch mechanism, both of which are typically implemented employing  
 15 Java Server Pages (JSP).

One advantage of employing JSP is that the use of JSP facilitates organization of a website as a state machine. In this manner, the logical organization of a website can be arranged in categories, for example: Controls, States and Transitions. Controls include a Java class of elements that manage the active elements of a page  
 20 such as render control text or interpret user's action with respect to a page. Examples of controls would be the management of a virtual button on a web page or login management that could include providing a number of dialog boxes containing text and a virtual button. States define a user's current location on the website (e.g., in a state machine), such as the web page that a user is presently viewing. States also  
 25 define the relationship of a user with respect to a web page being viewed. Transitions define the new state of a user and are a function of a user's interaction with a page. Specifically, a transition is defined by the user's current state and the actions taken by the user while in that state (e.g., the result of user operation on a control alters the user's state). Simply put, the user's new state is simply defined as the user's current

state, as modified by the transition selected. The transitions are located in a transition module that is responsible for all transitions.

Advantages of the state machine model of the website are that it has maintainability to facilitate update flow or pages very easily and per user state machine. service different users with maximum code reuse. It is also consistent in that most or all server logic is handled under the same paradigm. Login control, record and display controls cause transitions which update state. Typically, XML/XSL defines the state machine, page content and layout. It can also be compatible with the Wireless Application Protocol (WAP). This can extend existing web site to provide different state machines for WAP users.

Also included in the web server is a control layer and a module layer, both implemented in the Java programming language. The applications on the application server can be implemented, for example, in the Java programming language.

#### **An Example Architecture Supporting Product Comparison**

Fig. 4 is an example screen layout illustrating a visual representation of a "web page" 400 presented on display screen 24, for example, at the aforementioned URL address. Web page 400 includes, *inter alia*, a hypertext link 402. Employing a mouse (e.g.) mouse 46, a cursor 60 can be placed proximate to hypertext link 402, entitled "home" and a cursor event is effectuated (i.e., hypertext link 402 is activated).

Activating hypertext link 402 results in a visual representation of web page 400 being present on display screen 24. In this manner, hypertext link 402 connects to web page 400 by having the same rendered on display screen 24. Web page 400 includes a number of hypertext links 402-448, as well as a number of data entry fields 450-475. Also included on web page 400 is a hypertext link 480 for connecting to information promulgated over the Internet via e-mail, for example.

Hypertext links 414, 416, 418 and 434 connect to additional web pages that are in data communication with databases having information concerning products that are the subject of commercial transactions over the data network. For example, in

the case of new vehicle sales, hypertext link 414 allows the user to navigate to a page that includes information concerning various models of vehicles. Hypertext link 416 can include information concerning the ordering of a vehicle having a desired set of features. Hypertext link 418 can include information concerning various services  
 5 provided by carOrder.com. Hypertext link 434 can include information concerning financing of a vehicle to be purchased. Hypertext link 420, on the other hand, allows the user to navigate to a web page displaying information concerning vehicles already selected as being suitable for purchase by the user, as discussed in detail subsequently.

Hypertext links 406-412, 430 and 438-448 are provided to inform users of  
 10 certain information not germane to a description of embodiments of the present invention. For example, hypertext links 406, 410 and 448 connect to web pages that explain how to use the carOrder.com website. Hypertext links 408, 438, 440 and 442 provide company information concerning carOrder.com, such as summary of the company, services available and benefits thereof, press releases, jobs available, how to  
 15 contact the company and so on. Hypertext link 412 describes how privacy is maintained on the website. Hypertext link 402 connects to web page 400, and hypertext link 444 discusses the terms and conditions for gaining access to the website. Hypertext link 430 connects to web pages that allow a user to learn more about the carOrder.com website, the process of identifying, selecting and buying a car  
 20 using the carOrder.com website, and the like. Hypertext link 446 connects to web pages that allow other companies to affiliate themselves (and their websites) with the carOrder.com website.

A user gains access to the various functionalities provided by a website (e.g., web page 400) according to an embodiment of the present invention by placing cursor  
 25 60 proximate to hypertext link 428 and effectuating a cursor event, hereinafter referred to as activating or selecting a hypertext link. Before activating hypertext link 428, a user must either enter information corresponding to a preexisting account in data entry fields 470 and 475, or register for a new account by activating hypertext link 426. In this manner, a user can be associated with a storage area that is referred  
 30 to herein as a Virtual Garage®. To restrict access to the aforementioned storage area,



a code that corresponds to the user is associated with the Virtual Garage®. The  
 aforementioned code can include, for example, a user-name and a password. To gain  
 access to the user's Virtual Garage®, the user's user-name is entered in data-entry  
 region 470, with the appropriate password entered in data-entry region 475. As is  
 5 standard with most password security features, password data is not displayed in field  
 475.

Were a user attempting to gain access to the present invention for the first  
 time, hypertext link 426 would need to be activated to connect to an account  
 registration web page 500, shown in Fig. 5. Account registration web page 500  
 10 includes a number of data entry fields 502-526 in which the user enters their personal  
 information. Also included on web page 500 are data entry fields, such as data entry  
 fields 530, 532 and 540-558. Data entry fields 530 and 532 indicate the level of  
 restriction on dissemination of the information provided in data fields 502-526.  
 Information in data fields 540-558 indicates how the user became aware of the  
 15 carOrder.com website. A number of hypertext links not directly related to registration  
 of a user are also included on web page 500. For example, hypertext links 402-420  
 and 438-448 are the same as shown on web page 500, discussed above.

Upon acceptance of the registration data entered into the data entry fields 502-  
 526, the user's Virtual Garage® can be accessed by activating hypertext link 420.  
 20 Activation of hypertext link 420 connects to web page 600, shown in Fig. 6, that  
 includes information concerning products that the user is interested in purchasing. To  
 that end, web page 600 facilitates comparison price shopping by allowing a user to  
 store product-related information concerning multiple products and examine such  
 information. Information concerning various products (in this example) vehicles, can  
 25 be obtained by activating hypertext link 414, for example. Activating hypertext link  
 414 connects to web page 700, shown in Fig. 7A,

Fig. 7 includes, *inter alia*, a number of hypertext links that enable access to a  
 database of information related to products such as vehicles. For example, a hypertext  
 link 710 accesses a database concerning information organized by the make and

models of vehicles available. A Hypertext link 720 facilitates comparison of vehicles associated with the aforementioned database with other vehicles associated therewith or with information stored by the user in the Virtual Garage®. A Hypertext link 730 accesses the same database, but facilitates searching based upon the features associated with vehicles in the database. Other information is also available to the user, including views of car interiors (accessed using a hypertext link 740), a glossary of terms related to the given type of product offered (accessed using a hypertext link 742) and the sale of used products (accessed using a hypertext link 744).

Figs. 7, 8A and 8B are example screen layouts. Activating hypertext link 710 connects to a web page 800. Web page 800 includes a matrix 810 of hypertext links having the names of various states. Identifying the state in which a user will purchase a product allows the price of the given product (e.g., vehicle) to be accurately determined. In addition, a hypertext link 820 is present on web page 800 that connects to web page 700. Activating one of the hypertext links in matrix 810 connects to web pages having a listing of manufacturers.

Figs. 8B and 9 are example screen layouts illustrating the manufacturers for which products may be selected and the products available from the given manufacturer, respectively. Activating one of the hypertext links in matrix 810 connects to a web page such as web page 850. Web page 850 includes a matrix 860 of hypertext links having manufacturer names associated with various vehicle manufacturers. In addition, a hypertext link 870 is present on web page 850 that connects to web page 700. Activating one of the hypertext links in matrix 860 connects to web pages having a listing of models of cars fabricated by the manufacture whose name is associated with the hypertext link activated. For example, activating the hypertext link entitled "Jeep" connects to a web page 900. Web page 900 includes a brief description of different models of vehicles sold or manufactured under the "Jeep" trademark in a model selection area 910. In addition to the aforementioned descriptions, model selection area includes hypertext links 920, 922 and 924, which allow the user to learn more about the corresponding product. Also included in model selection area 910 are hypertext links 930, 932 and 934, which

allow the user to build a custom configuration of the given product according to the user's desires. In addition, web page 900 includes hypertext links 940 and 950.

Hypertext link 940 connects to web page 700 to allow a user to research other products. Hypertext link 950 connects to web page 850 to allow a user to view  
5 models of vehicles associated with a different manufacturer.

Referring to Figs. 9, 10, 11 and 12, hypertext links 920, 922 and 924 allow a user to obtain more specific information concerning particular vehicles sold/fabricated by the manufacturer. For example, hypertext link 922 connects to a web page 1000 that recites more detailed information concerning the Grand Cherokee model of Jeep  
10 vehicle in text region 1010 entitled "overview." Web page 1000 also includes various hypertext links that facilitate obtaining additional information concerning the Jeep Grand Cherokee.

These hypertext links include a hypertext link 1020 and a hypertext link 1030. Hypertext link 1020 connects to a web page (a web page 1100) containing  
15 information regarding performance characteristics of the Jeep Grand Cherokee. Web page 1100 is similar to web page 1000, excepting text region 1110 concerning the performance characteristics of the Jeep Grand Cherokee. Also similar are the presence of hypertext links 1120 and 1130, which connect to web page 1000 and a web page 1200, respectively. Hypertext link 1120 connects to web page 1000,  
20 returning the user to the overview of the Jeep Grand Cherokee. Hypertext link 1130 connects to web page 1200 containing information regarding the various features available on the Jeep Grand Cherokee.

Web page 1200 is similar to web page 1000, excepting recitations concerning the features available on the Jeep Grand Cherokee. Also, unlike web pages 1000 and  
25 1100, web page 1200 includes hypertext links 1220 and 1230 that allow connections to web pages 1000 and 1100, respectively. Hypertext link 1220, as with hypertext link 1120, connects to web page 1000, returning the user to the overview of the Jeep Grand Cherokee. Hypertext link 1230, as with hypertext link 1030, connects to web

page 1100, returning the user to the performance characteristics of the Jeep Grand Cherokee.

Web page 1000 also includes a hypertext link 1040 that allows a user to connect to web page 850 to choose information concerning vehicles associated with a different manufacturer. A Hypertext link 1050 allows a user to connect to web page 900 to obtain information related to a different model of the same manufacturer. Activating a hypertext link 1060 allows a user to choose different groups of features for the present model of vehicle that are provided by the manufacturer, referred to as option packages. In this manner, a user is able to "build" a vehicle model having the desired features.

Referring to Figs. 7, 13 and 14, from web page 700, a user can search for a particular vehicle based upon features desired. This functionality is presented to the user as a number of criteria. The user is able to select values, ranges or the like for one or more of these criteria, and in so doing, eliminate products (here, vehicles) from consideration. By performing such a selection process, the user is able to determine which products meet their requirements, as described by the selections made. To that end, hypertext link 730 can be activated, connecting to a web page 1300. Web page 1300 includes a number of hypertext links 1310-1360, which connect to various other web pages containing information concerning various features available on a list of vehicles recited in a column 1370. Web page 1300 also includes a data entry region 1380 having a number of data entry fields 1390(1)-(8). As depicted in Fig.13, each of data entry fields 1390(1)-(8) corresponds to a range of values. One or more of data entry fields 1390(1)-(8) can be selected to indicate the price of a vehicle in which a user is interested. It will be noted that a pair of numeric entries could be employed to allow a user to indicate the desired price range more exactly. The values, range of values or the like represented by data entry fields such as data entry fields 1390(1)-(8) typically depend on the feature or criteria being selected (e.g., price is typically selected as a range, while a body style is typically selected as a single value). In fact, depending on the feature, a number of such data entry fields may be selectable. For example, one or more of data entry fields 1390(1)-(8) can be selected to indicate the

price of a vehicle in which a user is interested. For a range of prices broader than a single range, multiple such ranges (multiple ones of data entry fields 1390(1)-(8)) can be selected. It will be noted that, when one or more price ranges are selected (or de-selected), column 1370 is typically configured to adjust the vehicles listed to reflect those vehicles within the currently-selected price range.

Activating hypertext link 1320 connects to a web page 1400. Web page 1400 is similar to web page 1300, excepting a data entry region 1410. A data entry region 1410 includes a number of data entry fields that correspond to characteristics of a vehicle, such as different options regarding body styles, drive trains and engine locations. It will be noted that, as such options are selected (or de-selected), column 1370 is preferably configured to adjust the vehicles listed to reflect those vehicles having the currently-selected options, and so make web page 1400 more user-friendly. Alternatively, the user can be required to indicate their desire that column 1370 be so updated.

Referring to Figs. 13 and 15, activating hypertext link 1330 connects to a web page 1500. Web page 1500 is similar to web page 1300, excepting the addition of a data entry region 1510. Data entry region 1510 includes a number of data entry fields that correspond to the characteristics of the engine associated with the desired vehicle. The aforementioned characteristics include the horsepower and the number of cylinders in the given engine. As noted, as such options are selected (or de-selected), column 1370 preferably adjusts the vehicles listed to reflect those vehicles having the currently-selected options.

Referring to Figs. 13 and 16, activating hypertext link 1340 connects to a web page 1600. Web page 1600 is similar to web page 1300, excepting the addition of a data entry region 1610 that includes a number of data entry fields that correspond to fuel economy of a vehicle. As a result, a user can select a vehicle based upon the fuel economy desired of the vehicle. For a desired fuel economy range broader than a single range, multiple such ranges (multiple ones of the data entry fields corresponding to the vehicle's desired fuel economy) can be selected. As noted, as

such fuel economy requirements are selected (or de-selected), column 1370 preferably adjusts the vehicles listed to reflect those vehicles having the desired fuel economy.

Referring to Figs. 13 and 17, activating hypertext link 1350 connects to web page 1700. Web 1700 is similar to web page 1300, excepting the addition of a data entry region 1710 which includes a number of data entry fields that correspond to the characteristics a vehicle's interior. The aforementioned characteristics can include the type of seats included with the vehicle, e.g., bench or bucket seats, as well as the upholstery thereof. Additionally, the type of sound system can be chosen from data entry regions in data entry region 1710. As noted, as such options are selected (or de-selected), column 1370 preferably adjusts the vehicles listed to reflect those vehicles having the currently-selected options.

Referring to Figs. 13 and 18, activating hypertext link 1360 connects to a web page 1800. Web page 1800 is similar to web page 1300, excepting the addition of a data entry region 1310 which includes a number of data entry fields that correspond to the safety characteristics a vehicle. This allows selecting vehicles based upon the safety restraint systems associated therewith, e.g., driver side airbag, integrated child safety seat, roadside assistance and the like. The aforementioned characteristics can include the type of seats included with the selected vehicle, (e.g., bench or bucket seats), upholstery and the like. Additionally, the type of sound system can be selected using data entry regions in data entry region 1310. As noted, as such options are selected (or de-selected), column 1370 preferably adjusts the vehicles listed to reflect those vehicles having the currently-selected options.

Fig. 19 is a flow diagram illustrating a needs analysis process 1900, in which a selection of cars is presented to the user based on analysis of that user's needs (i.e., a needs analysis). The process begins with the user's selection of the options, features or the like that are desired in the given product (e.g., vehicle) (step 1910). These options appear in web page 1300, for example, as data entry regions 1390(1)-(8). By selecting one or more of data entry regions 1390(1)-(8), the user is able to identify the user's desired price range, for example. If the user wishes to select a price range of

less than \$15,000, for example, the user selects data entry region 1390(1) (i.e., makes data entry region 1390(1) the subject of a cursor event, as described previously). The products meeting the given requirements (i.e., having the desired feature(s)) are then displayed (step 1920). As noted, this is preferably done as the user makes such  
5 selections. Thus, for example, a user can enter such requirements using the respective data entry regions of web pages 1300-1800. By doing so, the user narrows the field of possible product choices available to them, and so simplifies the task of selecting a product for purchase. The user is free to reconfigure the product, or may proceed to further configuration steps or to saving the product configuration (step 1930).

10 Having settled on a given product configuration (step 1930), the user may elect to further configure the product or store the configured product. If the user desires to further configure the product (step 1950), the user then navigates to web pages that allow such configuration (step 1960). These web pages can provide further detail, allowing the user to use the configured product as a starting point. The user  
15 can then select various features for the given product. Some of the features presented for selection by the user may not appear in the features presented by web pages 1300-1800, and so "fine tuning" of the product's configuration can be accomplished by the use of such web pages. Once the product has been configured to the user's satisfaction (either via, e.g., web pages 1300-1800, or through further configuration),  
20 the user can store the configured product, if desired (steps 1970 and 1980). Otherwise, the user can decide to exit without saving the configured product, or re-configure the product (step 1990).

It will be noted that an architecture such as that described herein allows such product configurations to be generated based on general criteria, such as price range,  
25 engine characteristics, and other such criteria. This is possible because certain corresponding configurations are typically defined for at least some of the products offered, in order to support the ability to generate configurations using such general criteria. It will also be noted that an architecture such as that described herein also allows the configuration information thus generated to be persisted to other functions  
30 within the architecture (e.g., a configuration engine that allows the user to vary certain

equipment features of the given vehicle, in the process of configuring a vehicle for evaluation, further configuration, storage (e.g., the storage of a configured vehicle in the user's Virtual Garage®), purchase and the like). Moreover, this architecture prevents the generation of product configurations that are not feasible, not allowed or offered by the seller or restricted in some other manner.

As noted, Fig. 19 depicts a flow diagram illustrating a selection process according to one embodiment of the present invention. It is appreciated that operations discussed herein may consist of directly entered commands by a computer system user or by steps executed by application specific hardware modules, but the preferred embodiment includes steps executed by software modules. The functionality of steps referred to herein may correspond to the functionality of modules or portions of modules.

The operations referred to herein may be modules or portions of modules (e.g., software, firmware or hardware modules). For example, although the described embodiment includes software modules and/or includes manually entered user commands, the various example modules may be application specific hardware modules. The software modules discussed herein may include script, batch or other executable files, or combinations and/or portions of such files. The software modules may include a computer program or subroutines thereof encoded on computer-readable media.

Additionally, those skilled in the art will recognize that the boundaries between modules are merely illustrative and alternative embodiments may merge modules or impose an alternative decomposition of functionality of modules. For example, the modules discussed herein may be decomposed into submodules to be executed as multiple computer processes, and, optionally, on multiple computers. Moreover, alternative embodiments may combine multiple instances of a particular module or submodule. Furthermore, those skilled in the art will recognize that the operations described in example embodiment are for illustration only. Operations



may be combined or the functionality of the operations may be distributed in additional operations in accordance with the invention.

Alternatively, such actions may be embodied in the structure of circuitry that implements such functionality, such as the micro-code of a complex instruction set computer (CISC), firmware programmed into programmable or  
5 erasable/programmable devices, the configuration of a field-programmable gate array (FPGA), the design of a gate array or full-custom application-specific integrated circuit (ASIC), or the like.

Each of the blocks of Fig. 19 may be executed by a module (e.g., a software  
10 module) or a portion of a module or a computer system user using, for example, a computer system such as the storage router previously mentioned, or a similar network element, as well as a computer system such as computer system 210. Thus, the above described method, the operations thereof and modules therefor may be executed on a computer system configured to execute the operations of the method  
15 and/or may be executed from computer-readable media. The method may be embodied in a machine-readable and/or computer-readable medium for configuring a computer system to execute the method. Thus, the software modules may be stored within and/or transmitted to a computer system memory to configure the computer system to perform the functions of the module.

Such a computer system normally processes information according to a  
20 program (a list of internally stored instructions such as a particular application program and/or an operating system) and produces resultant output information via I/O devices. A computer process typically includes an executing (running) program or portion of a program, current program values and state information, and the  
25 resources used by the operating system to manage the execution of the process. A parent process may spawn other, child processes to help perform the overall functionality of the parent process. Because the parent process specifically spawns the child processes to perform a portion of the overall functionality of the parent

process, the functions performed by child processes (and grandchild processes, etc.) may sometimes be described as being performed by the parent process.

Such a computer system typically includes multiple computer processes executing "concurrently." Often, a computer system includes a single processing unit which is capable of supporting many active processes alternately. Although multiple processes may appear to be executing concurrently, at any given point in time only one process is actually executed by the single processing unit. By rapidly changing the process executing, a computer system gives the appearance of concurrent process execution. The ability of a computer system to multiplex the computer system's resources among multiple processes in various stages of execution is called multitasking. Systems with multiple processing units, which by definition can support true concurrent processing, are called multiprocessing systems. Active processes are often referred to as executing concurrently when such processes are executed in a multitasking and/or a multiprocessing environment.

The software modules described herein may be received by such a computer system, for example, from computer readable media. The computer readable media may be permanently, removably or remotely coupled to the computer system. The computer readable media may non-exclusively include, for example, any number of the following: magnetic storage media including disk and tape storage media. optical storage media such as compact disk media (e.g., CD-ROM, CD-R, etc.) and digital video disk storage media. nonvolatile memory storage memory including semiconductor-based memory units such as FLASH memory, EEPROM, EPROM, ROM or application specific integrated circuits. volatile storage media including registers, buffers or caches, main memory, RAM, etc.. and data transmission media including computer network, point-to-point telecommunication, and carrier wave transmission media. In a UNIX-based embodiment, the software modules may be embodied in a file which may be a device, a terminal, a local or remote file, a socket, a network connection, a signal, or other expedient of communication or state change. Other new and various types of computer-readable media may be used to store and/or transmit the software modules discussed herein.

Fig. 20 is a block diagram illustrating a software architecture (specifically, a website architecture 2000) including the various layers and modules that can be used to create a website such as that depicted in Figs. 3-18 herein. Website architecture 2000 includes a scripts layer 2005, a controls layer 2010, a modules layer 2015, a services layer 2020 and a database layer 2025. Scripts layer 2005 includes a set of presentation scripts 2030, which may be implemented using a Java scripting language such as JSP, for example. Similarly, controls layer 2010 includes a set of presentation controls 2035. Presentation controls 2035 provide functionality that interoperates with presentation scripts 2030 to provide the user interface presented to a user accessing a website supported by a website architecture such as website architecture 2000. The functionality provided by presentation controls 2035 includes management of HTML controls, lists, links and buttons that provide the functionality presented by such a website.

Typically, controls have no state, except that which is needed for the actual control itself (e.g., identity). Modules layer 2015 includes a needs analysis module 2040, among other modules. In general, modules are used for tasks such as caching, cross-module communication and management of communications with the various services to which the given module belongs, which includes the management of object caching and the servicing of objects. As its name implies, needs analysis module 2040 provides the functionality necessary to analyze available products (e.g., vehicles) based on a user's indicated needs and so to support the functionality presented as web pages 1300, 1400, 1500, 1600, 1700 and 1800.

Services layer 2020 includes a filter service 2050 and a configuration service 2055. Services layer 2020 acts as an interface between modules layer 2015 and database layer 2025, and provides functionality such as data persistence and data preparation, allowing interaction with other functions within the architecture, such as remote resources. It will be noted that this persistence allows the configuration information generated during needs analysis to be made available to other functions within the architecture (e.g., further configuration or storage) without further user intervention (e.g., without requiring a user to re-enter the configuration information

thus generated). A services layer such as services layer 2020 also allows for the persistence of the respective information for each of the separate functionalities represented by each service therein, as well as control and management of the given service's data. Typically, a service has no state associated therewith.

5 In supporting the functionality provided by needs analysis module 2040, filter service 2050 and configuration service 2055 respond to requests for information from needs analysis module 2040 based on information provided by needs analysis module 2040. Such information is provided from the one or more databases of database layer 2025 (exemplified by a database 2060, which is also referred to herein as a product  
10 information database). Database layer 2025 (via database 2060) effects the actual storage of the data employed by the various functions provided by website architecture 2000. In providing such support, database 2060 provides services such as filter service 2050 and configuration service 2055 with the data necessary for these services to respond to requests from needs analysis module 2040.

15 For example, a user can send attribute information to needs analysis module 2040 as an attribute selection, using presentation scripts 2030 and presentation controls 2035. Needs analysis module 2040 then provides the attribute information (depicted in Fig. 20 as a set of attributes 2070) to filter service 2050, in order to allow a search of database 2060 based on the attribute information provided (i.e., attributes  
20 2070). Thus, a user is presented with web pages 1300-1800 by presentation scripts 2030 and presentation controls 2635, as well as the underlying functionality provided by needs analysis module 2040. When the user selects a given option (e.g., a price range of less than \$15,000) using a control (e.g., data entry field 1390(1)), information representing this selection is provided by needs analysis module 2040 to filter service  
25 2050 as attributes 2070. Filter service 2050 then uses this information to query database 2060 as to products (e.g., vehicles) that meet the user's requirements. Assuming one or more configurations match attributes 2070, the product identifiers for such vehicle configurations (exemplified by identifiers 2080) are passed from filter service 2050 to needs analysis module 2040, having been gleaned from database 2060  
30 by the filtering operation (i.e., search). This is the case where, for example, a user

wishes to identify vehicles meeting that user's stated criteria using the needs analysis techniques described herein. It will be noted that this and other requests discussed herein are in response to user actions presented via scripts layer 2005 (e.g., presentation scripts 2030) and controls layer 2010 (e.g., presentation controls 2035).

Alternatively, a user can send product identifier information to needs analysis module 2040 as a product identifier selection, using presentation scripts 2030 and presentation controls 2035. Needs analysis module 2040 then supplies the product identifier information (depicted in Fig. 20 as an identifier 2085) to configuration service 2055, in order to identify a list of configurations that correspond to the given identifier. This is the case, for example, where a user desires to see a list of possible configurations or equipment options available for a given vehicle. Once configuration service 2055 has received identifier 2085, configuration service 2055 accesses database 2060 to identify those parts (e.g., vehicle configurations) having an identifier matching identifier 2085. Once the search of database 2060 is complete, configuration service 2055 supplies a configuration list 2090 to needs analysis module 2040, for subsequent presentation to the user via scripts layer 2005 and controls layer 2010. Configuration list 2090 can include, for example, information regarding allowable combinations of features for a given product, a list of features available for a given product, or the like. Configuration service 2055 can thus be used, for example, to determine if a vehicle configuration meeting a user's needs exists or if a given vehicle is available with a given feature.

Fig. 21 is a block diagram illustrating the creation of a database such as database 2060 (referred to herein as a database creation architecture 2100). This is, in effect, a pre-processing step in which a (potentially large) number of possible configurations are enumerated for inclusion in database 2060. In the case of vehicles, as with many commercially-available products, a relatively large number of permutations can exist when considering various features that may be available for such products. With a large number of features, and so possible combinations, the problem of quickly presenting a given configuration becomes one of a combinatorial nature. To avoid (or at least lessen) such effects, embodiments of the present

invention employ a pre-processing operation in creating database 2060. Thus, for each product (e.g., vehicle) included in database 2060, a separate database is used. Such databases are exemplified in Fig. 21 by product databases 2110(1)-(N). Product databases 2110(1)-(N) include information regarding the features of each such product. For example, each one of product databases 2110(1)-(N) can include information regarding the available equipment options for a given make and model of a vehicle. Rules governing allowable combinations of such features can also be included in product databases 2110(1)-(N). For example, in the context of vehicle equipment configurations, certain combinations of attributes are either not possible or not allowable. This might be the case, for example, where two equipment options where either mutually exclusive (e.g., cloth upholstery and leather upholstery) or where the manufacturer does not permit such combinations (e.g., a sun-roof in a base model).

With regard to the present invention, the ability to discern allowable feature combinations is important because such capabilities preclude the presentation of products that fail to meet all the user's requirements in a single model. For example, a product, such as a vehicle, may be offered in a number of models. One of the product's models may meet one user requirement, while another of the product's models may meet another of the user's requirements. Thus, without knowledge of allowable feature combinations, a needs analysis process could present a product as meeting both requirements, while in fact a model meeting both requirements would not exist. By controlling which feature combinations are allowable, embodiments of the present invention avoid generating spurious results when performing needs analysis.

Such rules can also enumerate the precedence with which product features are satisfied when generating a comparable product (e.g., vehicle type, price, engine characteristics, trim features and so on). Other such situations will be readily apparent to one of skill in the art. Thus, a database manager 2120 is employed to process the information in product databases 2110(1)-(N) into (allowable) configurations for entry into database 2060. Moreover, maintenance of a given product's information is

simplified. This is because the product's information is more accessible (because the product information is stored separately from the configuration and attribute information stored in database 2060) and quickly accessed (due to the relatively small database in which the product's information is stored).

Database manager 2120 can be configured to perform this translation in a number of different ways. For example, database manager 2120 can generate the set of feature combinations that users are likely to desire from the information in product databases 2110(1)-(N) for entry into database 2060. Alternatively, database manager 2120 can be configured to generate all possible product configurations from product databases 2110(1)-(N), for entry into database 2060. Another possible scenario is to have research manager 2120 generate a minimal set of configurations from product databases 2110(1)-(N), and then add configurations dynamically to database 2060 as such configurations are requested by users of the websites employing website architecture 2000.

Fig. 22 illustrates an example of a database structure that can be used in database 2060. This database structure (exemplified in Fig. 22 by a database structure 2200) includes information regarding various product configurations (e.g., vehicle equipment configurations) and the attributes that make up such configurations, for example. This information can be held in a configuration table 2210 and an attribute table 2220, respectively, which are simply example representations of structures which may be used to maintain such information. Configuration table 2210 includes identifier fields 2230(1)-(N), configuration fields 2240(1)-(N) and intersection fields 2250(1)-(N). A record in configuration table 2210 (exemplified by one of configuration table records 2255(1)-(N)) includes a corresponding one of each of identifier fields 2230(1)-(N), configuration fields 2240(1)-(N) and intersection fields 2250(1)-(N). Each of configuration table records 2255(1)-(N) corresponds to one of the configurations created by research manager 2120 from a corresponding one of product data bases 2110(1)-(N). In other words, the configuration represented by the corresponding one of configuration fields 2240(1)-(N) is generated from the product database corresponding to the given product using the information and rules contained

in the corresponding one of product databases 2110(1)-(N). Thus, each configuration generated from product databases 2110(1)-(N) by research manager 2120 appears in configuration table 2210 as one of configuration fields 2240(1)-(N) and can be accessed as such using the corresponding one of identifier fields 2230(1)-(N) for example. Also provided as part of configuration table 2210 are intersection fields 2250(1)-(N). Intersection fields 2250(1)-(N) allow access to attribute table 2220 from a given one of configuration table records 2255(1)-(N).

Attribute table 2220 includes intersection fields 2265(1)-(N) and a number of attribute fields (depicted in Fig. 22 by attribute fields 2270(1,1)-(N,N)) that are broken up into a set of records (exemplified by attribute table records 2280(1)-(N)). While configuration table 2210 maintains information regarding various configurations of products and their features, attribute table 2220 maintains corresponding information with regard to the actual attributes of the product. Intersection fields 2265(1)-(N) allow access to attribute table 2220 from a given one of configuration table records 2255(1)-(N). Records in configuration table 2210 and attribute table 2220 reference one another via references such as a reference 2260. For example, configuration table record 2255(1) is referenced to attribute table record 2280(3) using information held in intersection field 2250(1), and attribute table record 2280(3) is referenced to configuration table record 2255(1) using information held in intersection field 2265(3). This cross-referencing is depicted in Fig. 22 by reference 2260. Thus, searching configurations using attributes, or *vice versa* (searching attributes using configurations (e.g., configuration IDs)) can be performed.

It will be noted that the references between configuration table 2210 and attribute table 2220 are bi-directional (e.g., reference 2260). This allows the a process according to embodiments of the present invention to be performed in support of the searching of database 2060, based on the user's needs (i.e., needs analysis or sorting products by features). Such functionality appears in Fig. 20 as filter service 2050, which receives attributes 2070 from needs analysis module 2040 and uses these attributes to access information in database 2060, as follows. Upon receiving attributes 2070, filter service 2050 uses the attributes therein to access attribute table



2220 by matching (in some way) one or more of attribute fields 2270(1,1)-(N,N) in order to identify those of attribute table records 2280(1)-(N) which are desired. Once the desired attribute table records have been identified, each corresponding reference is used to access a given one of configuration table records 2255(1)-(N). Having  
5 identified desired ones of configuration table records 2255(1)-(N), information stored in each one of identifier fields 2230(1)-(N) corresponding to the desired set of attributes can then be provided by filter service 2050 to needs analysis module 2040 (e.g., identifier 2080). This corresponds to filter service 2050 passing identifier 2080 to needs analysis module 2040. Using such an approach, in the situation where a  
10 given product (e.g., a vehicle listed in column 1370 of web pages 1300-1800) represents multiple configurations (e.g., models), the product's listing implies that at least one model of the given product exists that meets the user's requirements.

Another permutation supported by database structure 2200 is depicted in Fig. 20 in the provision of identifier 2085 by needs analysis module 2040 to configuration  
15 service 2055. In this situation, identifier 2085 is used by configuration service 2055 in accessing database 2060 in order to generate a list of configurations matching identifier 2085 (depicted in Fig. 20 in the response by parts service 2055 as configuration list 2090). In this scenario, identifier 2085 is used to identify one or more of configuration table records 2255(1)-(N) corresponding to the desired  
20 configurations by matching identifier 2085 to one or more corresponding ones of identifier fields 2230(1)-(N), and so to access corresponding ones of configuration fields 2240(1)-(N). Once the desired configuration fields have been identified, this information is passed by configuration service 2055 to needs analysis module 2040 as configuration list 2090.

25 Fig. 23 is a flow diagram illustrating an example of the operation of website architecture 2000, when used to generate product configurations from selected attributes (as depicted in Figs. 13-18). The process begins with the user entering attribute selections (e.g., such as those described in relation to Figs. 13-18) (step 2300). This functionality is provided by presentation scripts 2030 and presentation  
30 controls 2035, for example, or other, comparable software residing in scripts layer

2005 and controls layer 2010. This attribute information is then made available to needs analysis module 2040 (step 2310). Next, the attribute information is sent from needs analysis module 2040 to filter service 2050 as attributes 2070 (step 2320). Filter service 2050 then queries database 2060 using the attribute information supplied (step 2330). Next, filter service 2050 returns information to needs analysis module 2040 regarding the product(s) identified as having the desired attributes (step 2340). Needs analysis module 2040 then supplies the information regarding product(s) having the desired attributes to higher levels for display (e.g., presentation scripts 2030 and presentation controls 2035, or other such functionality provided by scripts layer 2005 and controls layer 2010) (step 2350).

Fig. 24 is a flow diagram illustrating an example of the operation of website architecture 2000, when used to identify product features (i.e., attributes). The process begins with the user entering one or more product or product configuration selections (step 2400). This functionality is provided by presentation scripts 2030 and presentation controls 2035, for example, or other, comparable software residing in scripts layer 2005 and controls layer 2010. This product identifier information is then made available to needs analysis module 2040 (step 2410). Next, the product identifier information is sent from needs analysis module 2040 to configuration service 2055 as identifier 2085 (step 2420). Configuration service 2055 then queries database 2060 using the product identifier information supplied (step 2430). Next, configuration service 2055 returns information to needs analysis module 2040 regarding the product's (or products') configuration(s) identified as having the selected product identifier(s) (step 2440). Needs analysis module 2040 then supplies the information regarding these configurations to higher levels for display (e.g., presentation scripts 2030 and presentation controls 2035, or other such functionality provided by scripts layer 2005 and controls layer 2010) (step 2450).

Fig. 25 is a block diagram illustrating an example of website 2500 employing website architecture 2000. In addition to needs analysis module 2040, website 2500 includes a configuration engine 2510 and a Virtual Garage® 2520. Configuration engine 2510 allows a user to configure a vehicle (typically are a finer level of detail

than that of needs analysis module 2040), while Virtual Garage® 2520 permits the user to store a configured vehicle. In addition to storing a vehicle configured using configuration engine 2510, Virtual Garage® 2520 is also configured to accept such information from needs analysis module 2040. Moreover, Virtual Garage® 2520 can accept such information without user intervention (e.g., requiring a user to re-enter configuration information regarding the given vehicle). This ability is supported by the data persistence discussed previously, and so entails the functionalities provided by services layer 2020. Once stored in the user's Virtual Garage® (e.g., Virtual Garage® 2520), the configured vehicle can be loaded into configuration engine 2510 for review and/or re-configuration.

In a similar fashion, a configured vehicle can be transferred between needs analysis module 2040 and configuration engine 2510. For example, a vehicle configured in needs analysis module 2040 (e.g., by a user navigating to one or more of web pages 1300-1800, or by some other method) can be transferred to configuration engine 2510. This allows a user to further configure a vehicle configuration, identified as desirable using needs analysis module 2040, using configuration engine 2510. The vehicle configuration, once "fine-tuned" using needs analysis module 2040, can then be stored in Virtual Garage® 2520. This process is simplified by the persistence provided by services layer 2020, which again obviates the need for a user to re-enter configuration information regarding the given vehicle.

The ability to persist a vehicle configuration can be supported by a number of features that are preferably included website architecture 2000. First, persistence is supported by the inclusion, in a given configuration of a vehicle, of all information necessary to elucidate all relevant equipment features selected by a user (regardless of the method used to indicate such selections (e.g., by comparison, configuration or by another method). This also provides support for ensuring the existence of a given configuration, which avoids a user's selection of incompatible or non-existent feature combinations, as noted earlier. Moreover, it is preferable that a Virtual Garage® such as Virtual Garage® 2520 maintain such information, to avoid the need for a user to re-enter such information upon exit from and re-entry to the website, and to simplify and

speed the use of website 2500. The inclusion of all relevant information (e.g., all necessary configuration information, including all user-selectable information) also allows such configurations to easily be passed from needs analysis module 2040 to configuration engine 2510 and Virtual Garage® 2520, again obviating the need for re-entry by the user.

The configuration of database 2060 also enhances the transfer of vehicle configurations between needs analysis module 2040 and configuration engine 2510, and between needs analysis module 2040 and Virtual Garage® 2520. By providing the ability to generate a configuration list from an identifier, using configuration service 2055, website architecture 2000 allows needs analysis module 2040 to quickly and easily generate a configured vehicle. This can be done, for example, by identifying the desired vehicle configuration using some or all of the information depicted in web pages 1300-1800. Once generated, this configuration can then easily be passed to configuration engine 2510 or Virtual Garage® 2520.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims.